DESIGN INTERNSHIP (DI-41)

FOR

MAVEN
SILICON
Centre of Excellence in VLSI

AHB TO APB BRIDGE DESIGN

PROJECT REPORT

DONE BY -

ANANYA JAIN

Aim of the Project:

This project aims to design a bridge between the Advanced High-performance bridge (AHB) and the Advanced Peripheral bus (APB) which are two different distinct buses defined within the Advanced Bus Microcontroller Architecture (AMBA) Specification. The bridge actually allows Communication between High Performance devices and Low Peripheral Devices.

Objectives of the Project:

1) Study the Top Module Block Diagram
2) Write Verilog code for different modules of the APB2AHB Bridge
3) Verify the same using a Test Bench.
4) Generate relevant output waveforms for better visualization and understanding.

Protocol Of Project->

1. **Initialization**:
    - The AHB to APB Bridge is initialized with necessary configuration parameters such as clock frequency, address mappings, and control settings.
2. **AHB Slave Configuration**:
    - The AHB to APB Bridge is configured as an AHB slave with appropriate address mapping and control signals.
3. **APB Master Configuration**:
    - The AHB to APB Bridge is configured as an APB master with appropriate address mapping and control signals.
4. **Transfer Request Handling**:
    - Upon receiving a read or write request from the AHB master, the AHB to APB Bridge decodes the address and control signals.
    - For read requests:
        - The AHB to APB Bridge converts the AHB read request into an equivalent APB read request.
        - The corresponding APB read transaction is initiated.

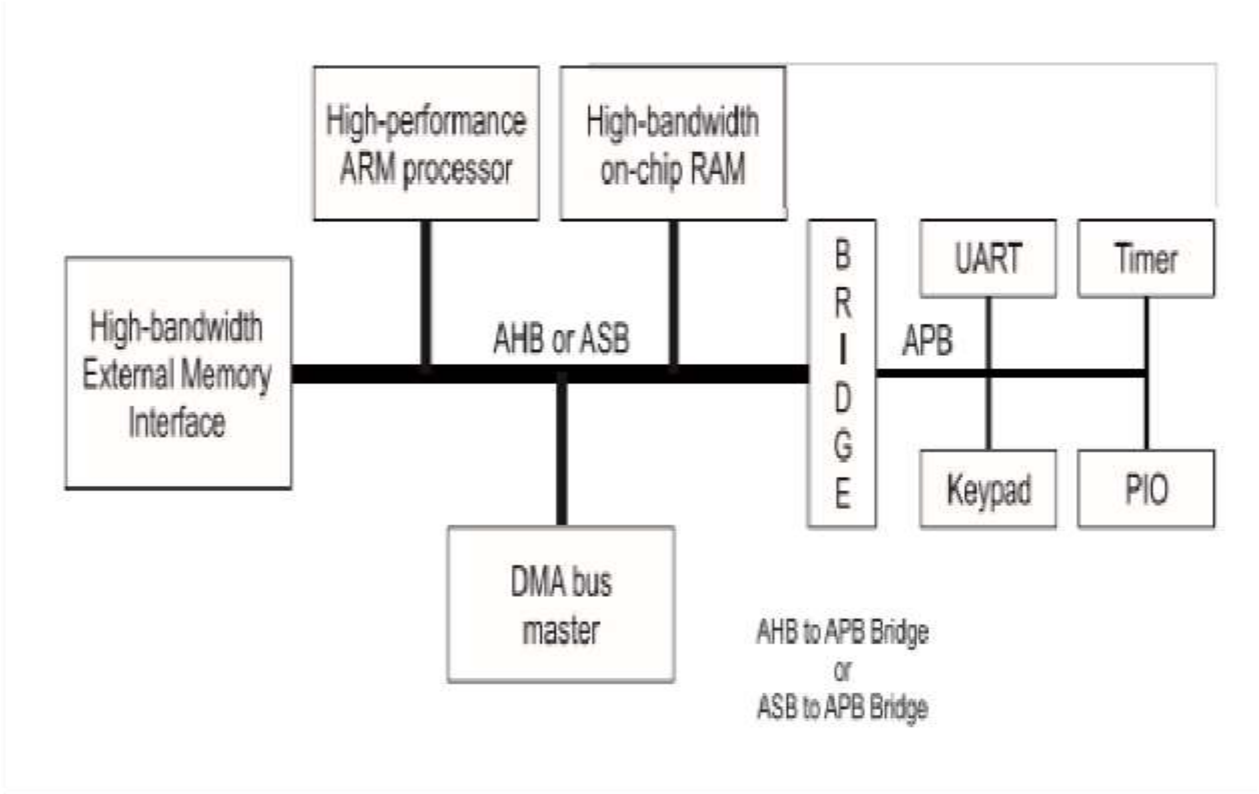
        - Data from the APB slave is read and transferred back to the AHB master.
    - For write requests:
        - The AHB to APB Bridge converts the AHB write request into an equivalent APB write request.
        - The data is written to the APB slave.
5. **Response Handling**:
    - Upon completion of the APB read or write transaction, the AHB to APB Bridge captures the response.
    - For read transactions, the data read from the APB slave is transferred back to the AHB master.
    - For write transactions, acknowledgment of successful write is sent to the AHB master.
6. **Error Handling**:
    - If any error occurs during the transfer, such as bus contention or timeout, appropriate error handling mechanisms are activated.
    - Error responses are generated and communicated to the AHB master.

7. **Interrupt Handling**:
    - The AHB to APB Bridge may support interrupt generation based on certain conditions.
    - Interrupt signals are generated and communicated to the AHB master as required.
8. **Power Management**:
    - The AHB to APB Bridge may implement power management features to optimize energy consumption.
    - Idle or low activity states trigger power-saving modes while maintaining responsiveness to incoming requests.
9. **Clock Synchronization**:
    - Clock domains between AHB and APB may differ. The bridge should handle clock domain crossing efficiently to ensure reliable data transfer.
10. **Protocol Compliance**:
    - The AHB to APB Bridge adheres to the specifications of both the AHB and APB protocols.
    - Timing requirements, signal levels, and protocol constraints are strictly followed to ensure compatibility with AHB and APB peripherals.

11. **Reset Handling**:
    - Proper reset mechanisms are implemented to initialize the bridge into a known state upon system reset or initialization.
12. **Testing and Verification**:
    - Rigorous testing and verification procedures are conducted to ensure the functionality, reliability, and performance of the AHB to APB Bridge under various operating conditions and corner cases.

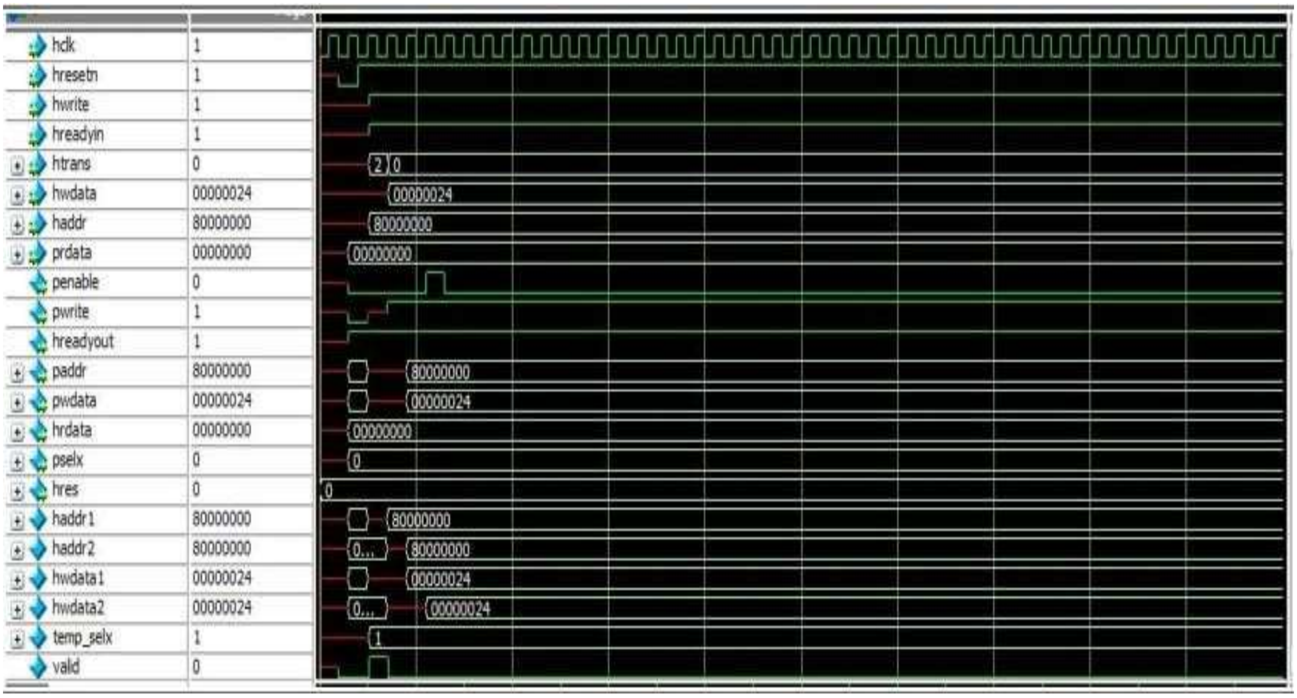
This protocol outlines the essential steps and considerations for the operation of an AHB to APB Bridge, ensuring seamless communication between high-speed AHB and low-power APB peripherals.

AMBA Based Microcontroller:


High-performance ARM processor
High-bandwidth on-chip RAM
High-bandwidth External Memory Interface
AHB or ASB
B R I D G E
UART
Timer
APB
Keypad
PIO
DMA bus master
AHB to APB Bridge or ASB to APB Bridge

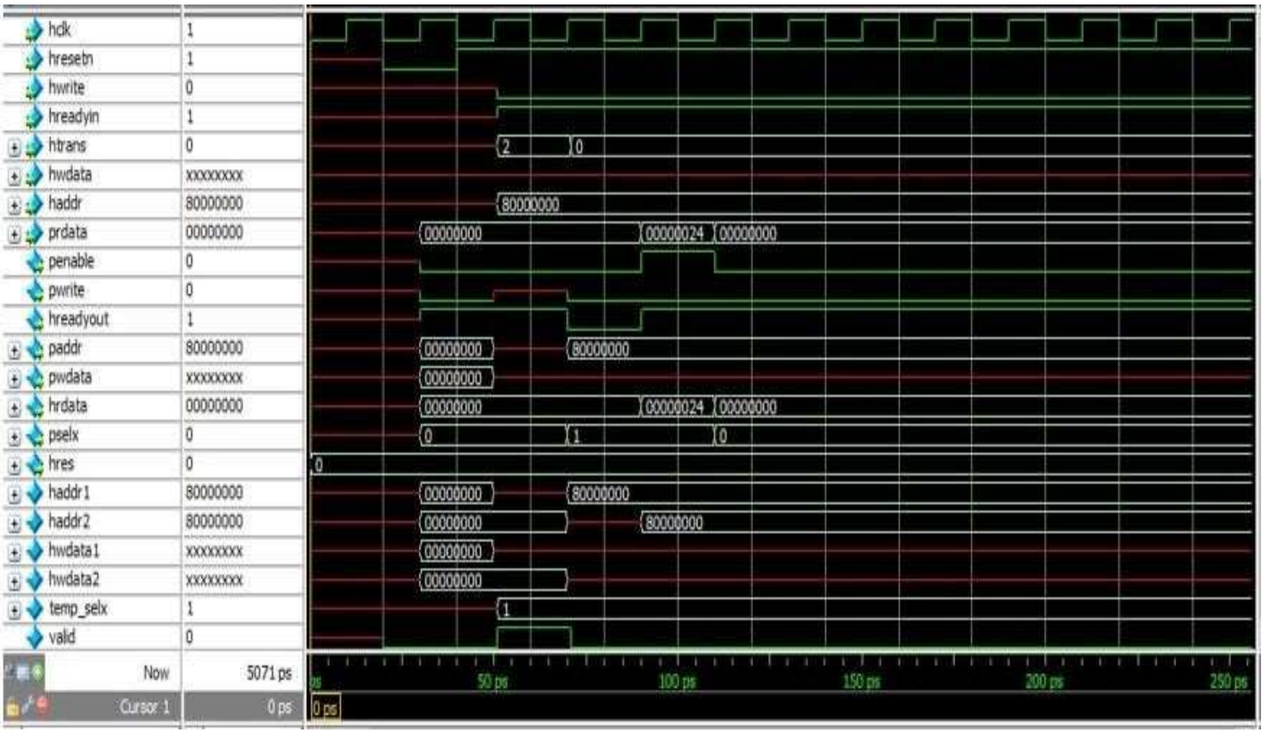
Simulation Waveform for output:

SINGLE WRITE:



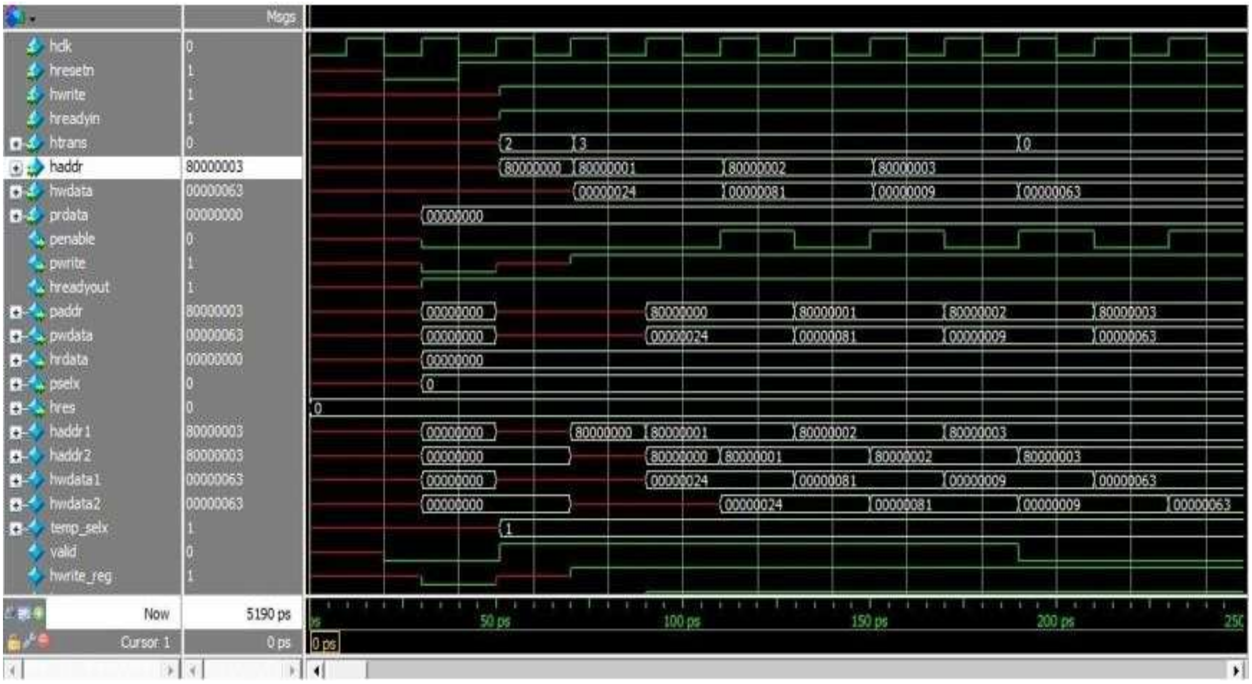
hclk	1
hresetn	1
hwrite	1
hreadyin	1
htrans	0
hwdata	00000024
haddr	80000000
prdata	00000000
penable	0
pwrite	1
hreadyout	1
paddr	80000000
pwdata	00000024
hrdata	00000000
pselx	0
hres	0
haddr1	80000000
haddr2	80000000
hwdata1	00000024
hwdata2	00000024
temp_selx	1
valid	0

SINGLE READ:

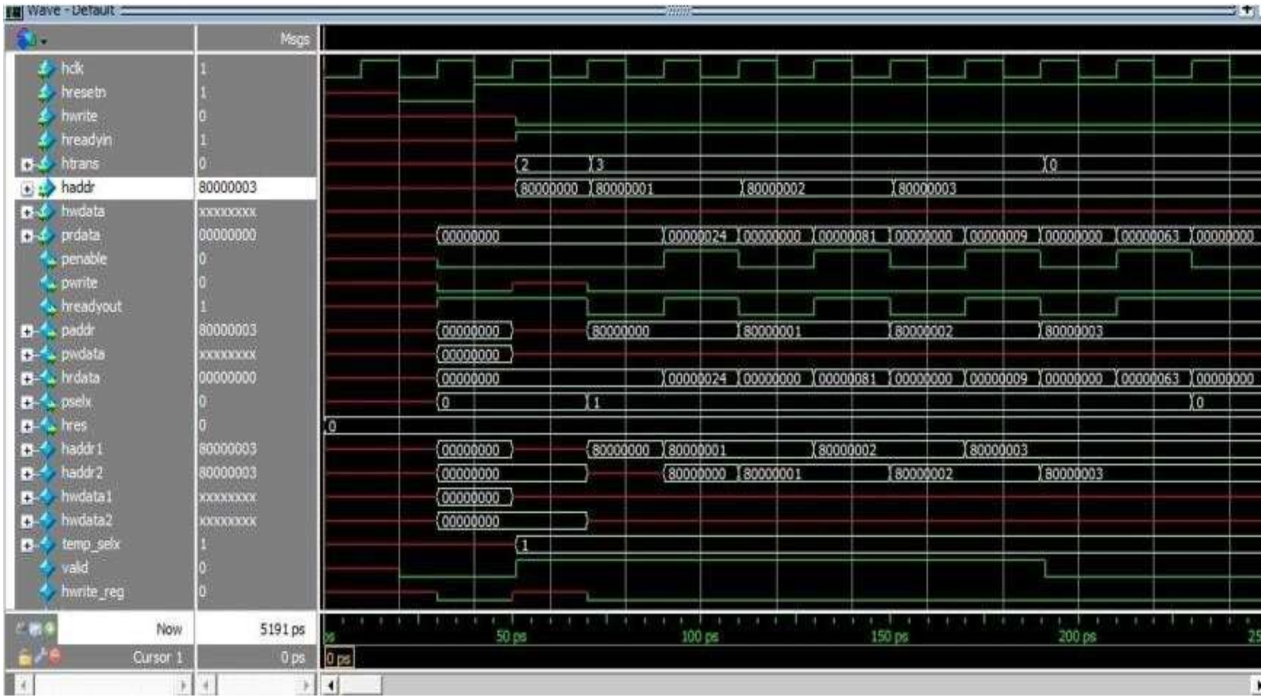


hclk	1
hresetn	1
hwrite	0
hreadyin	1
htrans	0
hwdata	xxxxxxxx
haddr	80000000
prdata	00000000
penable	0
pwrite	0
hreadyout	1
paddr	80000000
pwdata	xxxxxxxx
hrdata	00000000
pselx	0
hres	0
haddr1	80000000
haddr2	80000000
hwdata1	xxxxxxxx
hwdata2	xxxxxxxx
temp_selx	1
valid	0
Now	5071 ps
Cursor 1	0 ps

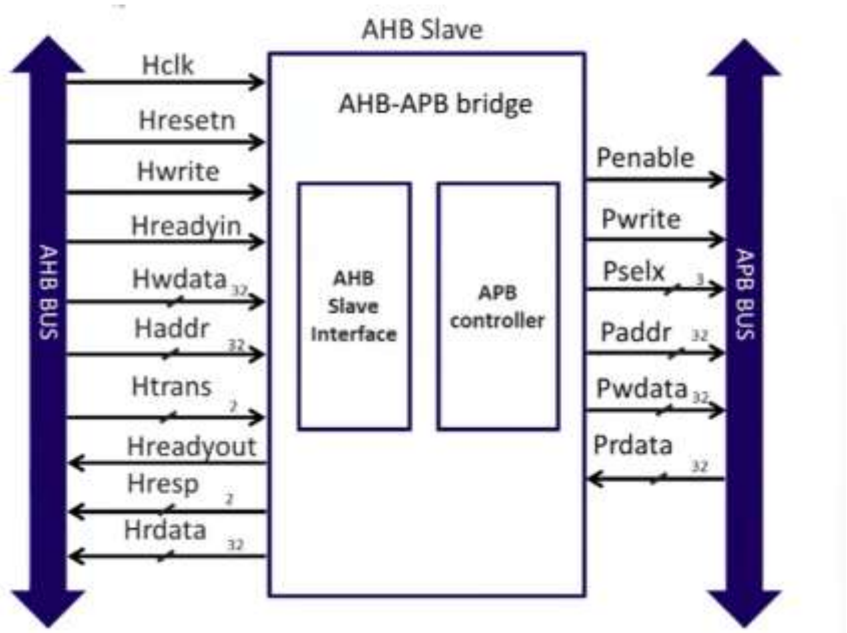
BURST WRITE:



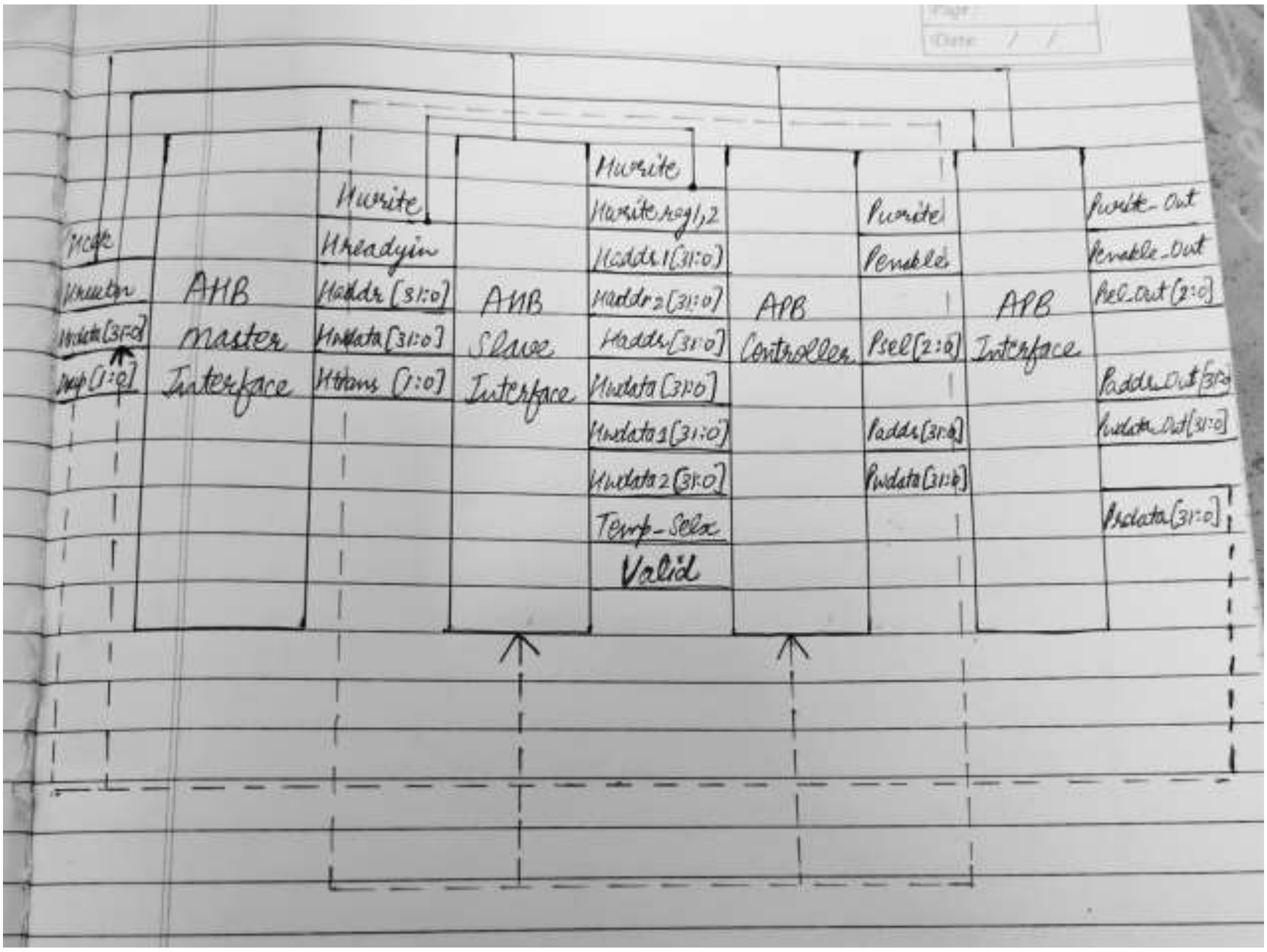
BURST READ:



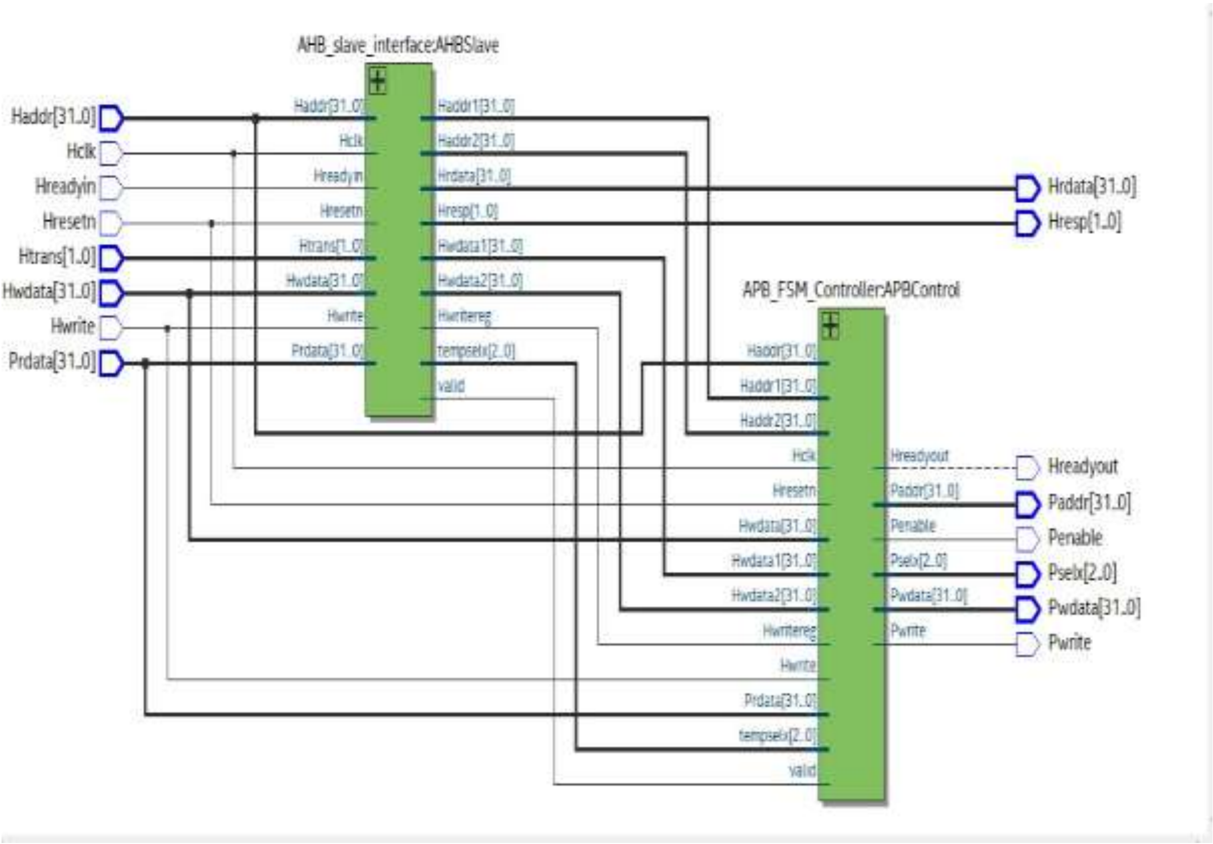
Block Diagram:



AHB-APB Bridge Interface Architecture:



Synthesis output:



AHB_slave_interface:AHBSlave
Haddr[31..0]
Hclk
Hreadyin
Hresetn
Htrans[1..0]
Hwdata[31..0]
Hwrite
Prdata[31..0]
Haddr1[31..0]
Haddr2[31..0]
Hrdata[31..0]
Hresp[1..0]
Hwdata1[31..0]
Hwdata2[31..0]
Hwritereg
tempselx[2..0]
valid
Hrdata[31..0]
Hresp[1..0]
APB_FSM_Controller:APBControl
Haddr[31..0]
Haddr1[31..0]
Haddr2[31..0]
Hclk
Hresetn
Hwdata[31..0]
Hwdata1[31..0]
Hwdata2[31..0]
Hwritereg
Hwrite
Prdata[31..0]
tempselx[2..0]
valid
Hreadyout
Paddr[31..0]
Penable
Pselx[2..0]
Pwdata[31..0]
Pwrite
Hreadyout
Paddr[31..0]
Penable
Pselx[2..0]
Pwdata[31..0]
Pwrite